REMARKS

Applicant amended independent claim 13 to include a feature, similar to the features recited in independent claims 26 and 31, that individual ones of the multiple processing engines of applicant's processor include corresponding arbiters to select threads for execution. After this amendment, claims 13-35 are pending. Claims 13, 26 and 31 are independent.

The Examiner rejected claims 13-21 and 24 under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 6,295,600 to Parady. Additionally, the examiner rejected claims 22 and 23 under 35 U.S.C. §103(a) as being unpatentable over Parady. The Examiner also rejected claims 26-35 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,085,215 to Ramakrishnan, in view of Parady.

Specifically, with respect to applicant's feature pertaining to the multi-threaded processing engines, and individual ones of the engines including an arbiter, which was recited in claim 26 and appears in amended independent claim 13, the examiner admitted that "Ramakrishnan did not specify (claims 26, 31) the internal configuration of the network processor that processed the multiple threads in real-time," and "Ramakrishnan did not specify (claim 26) that individual ones of the engines including an arbiter to select a thread to execute" (Office Action, paragraphs 25 and 27 on page 7). The examiner, however, contends that Parady discloses these features. Applicant respectfully disagrees.

Applicant's independent claim 13 recites "[a] method, comprising: at a processing engine within a processor having multiple processing engines, individual ones of the engines including corresponding arbiters to select threads for executions."

In contrast, Parady describes apparatus for switching between threads of a program in response to a long-latency event (Abstract). Particularly, Parady's apparatus includes an instruction decode unit 14 that decodes and provides instructions to four instruction buffers 102, 104, 106 and 108, corresponding to separate threads supported by Parady's apparatus (FIG. 3, col. 3, lines 44-52). A dispatch unit 28 provides instructions from the buffers to an execution unit 41 that includes functional units 32-46 (FIGS. 1 and 3, col. 3, lines 19-23). Parady further describes that its apparatus also includes thread-switching logic 112 that gives the hardware

thread-switching capabilities (FIG. 3, col. 3, lines 66-67). At no point, however, does Parady describe that its apparatus, in all its various embodiments as shown in FIGS. 3, 5 and 6, includes more than a single thread-switching unit. Indeed, since Parady's apparatus includes only a single execution unit 41 capable of processing only one thread at a time, having more than one thread-switching logic unit would be entirely unnecessary. Accordingly, Parady neither discloses nor suggests "at a processing engine within a processor having multiple processing engines, individual ones of the engines including corresponding arbiters to select threads for execution," as required by applicant's independent claim 13.

The examiner, however, argues that applicant's features of a processor with multiple processing engines and the individual ones of the engines including corresponding arbiters to select threads for executions are disclosed in Parady's in that:

> Parady, however, taught a processor comprising multiple multithreaded processing engines (32,34,36,38,40,42,44,46)[note multiple threads for integer thread in integer register files in integer execution logic and multiple threads for floating point threads stored in FP register files in floating point execution logic in figure 3 and figure 5 (e.g., see col. 5, lines 16-28)], (Office Action, Paragraph 25, Page 7).
>
> ...
>
> Parady taught grouping of the threads into a integer thread group stored in integer register files (48,158) and a floating point thread group stored in floating point register files (50,172) that are respectively executed on the integer execution logic and floating point execution logic (e.g., see figs. 3,5 and col. 5, lines 16-28). One of ordinary skill would have been motivated to include individual thread switching logic within the processing engines in at least one implementation of the Ramakrishnan and Parady teachings at least to provide the system with the ability to efficiently perform thread switching when only one floating point unit and one integer unit were used for the plurality of grouped threads such as when the other execution logic was not operational or when to reduce system cost only one integer and one floating point unit was employed.

Applicant respectfully disagrees with this characterization.

As described in Parady, the units 32-46 are functional units of a single processor 10, and include "the load/store unit 32 and the two integer ALU units 34 and 36, share a set of integer registers 48. Floating-point registers 50 are shared by floating point units 38, 40 and 42 and graphical units 44 and 46" (FIG. 1, col. 3, lines 27-30). Thus, the functional units 32, 34, 36, 38, 40, 42, 44 and 46 are modules within a processor configured to perform specific functions in the course of processing instructions received by the processor (e.g., the Integer ALU 36 would

presumably perform ALU operations on integer operands specified in the received instructions). Thus, the functional units 32, 34, 36, 38, 40, 42, 44 and 46 do not correspond to a processor having multiple processing engines.

The examiner contends that: "[o]ne of ordinary skill would have been motivated to include individual thread switching logic within the processing engines, ..." Applicant disagrees that this is sufficient to establish a *prima facie* case of obviousness. In order to establish a *prima facie* case, the references must teach or suggest all the claim limitations. However, by the examiner's own admission the combination of Ramakrishnan and Parady disclose only a single thread switching logic. Thus the combination of the cited references fails to disclose "individual ones of the engines including corresponding arbiters to select threads for executions," as required by applicant's independent claim 13. The examiner has not provided any basis upon which one of ordinary skill in the art would be motivated to modify the alleged combination of references.

Moreover, applicant contends that one of ordinary skill in the art would not be motivated to include individual thread switching logic within the alleged processing engines in Parady since none of the units 32, 34, 36, 38, 40, 42, 44 and 46 is a processing engine within the meaning of claim 13.

For the foregoing reasons, applicant submits that neither Ramakrishnan nor Parady discloses or suggest, alone or in combination, applicant's claim 13 feature of "a processing engine within a processor having multiple processing engines, individual ones of the engines including corresponding arbiters to select threads for execution." Applicant's independent claim 13 is therefore patentable over the cited reference.

Claims 14-25 depend from independent claim 13, and are therefore patentable for at least the same reasons as independent claim 13.

Independent claims 26 and 31 recite "multiple, multi-threaded processing engines, individual ones of the engines including an arbiter to select a thread to execute," or similar language. For at least similar reasons as those provided with respect to independent claim 13, at least this feature is patentable over the cited art. Claims 27-30 depend from independent claim 26 and are therefore patentable for at least the same reasons as independent claim 26. Claims 32-

35 depend from independent claim 31 and are therefore patentable for at least the same reasons as independent claim 31.

In addition, as noted above, the examiner rejected claim 22 under 35 U.S.C. §103(a) as being unpatentable over Parady. Specifically, the examiner stated:

> 19. As per claims 22,23 Parady did not expressly detail at least one instruction identifies the signal. However since Parady taught a system where a L2 cache miss signal is received by thread switching logic and signals from thread switching logic are sent via a common line to the plural threads (e.g., see fig. 3) There it would have been obvious to one of ordinary skill that the instruction in a particular thread that was executing would have identified its signal since each thread is sent each thread switching logic signal (otherwise all the threads would not be independent). As the claim is understood the instruction identifies the signal by instructing the apparatus to determine whether the proper signal was received acknowledging receipt of the request or indicating the thread was to be switched or not due to long latency in processing the request, and this was taught by Parady as discussed above. (Office Action, paragraph 19, pages 5-6)

Applicant respectfully disagrees.

Applicant's claim 22 recites "[t]he method of claim 13, wherein the at least one instruction identifies the signal." Using an instruction to identify a signal, corresponding to a shared resource that may be accessed by one or more threads of one or more microengines, enables synchronization of task completion and execution swapping of the multiple threads operating on applicant's multiple microengines (see, for example, first paragraph of page 7 of the originally filed application).

In contrast, Parady describes:

> **Thread switching logic 112 is provided to give a hardware thread-switching capability. The indication that a thread switch is required is provided on a line 114 providing an L2-miss indication from cache control/system interface 22 of FIG. 1. Upon such an indication, a switch to the next thread will be performed, using, in one embodiment, the next thread pointer on line 116. The next thread pointer is 2 bits indicating the next thread from an instruction which caused the cache miss. (col. 3, line 66, to col. 4, line 7)**

Thus, Parady merely describes that a cache miss would trigger a thread swapping operation, presumably because the thread associated with the instruction that resulted in the cache miss will now have to wait until the data the instruction requested is retrieved from a slower memory module. But at no point does Parady describe that any instruction, including the instruction whose execution resulted in the cache miss, identifies a signal that when generated

will cause the thread associated with that instruction to be swapped back to resume execution. Indeed, since Parady's system includes a single processor that executes one thread at a time, Parady's does not have to address the synchronization problems that arise with the use of multiple threads executing on multiple microengines. Therefore, Parady does not have to identify in the instructions executing on its apparatus signals corresponding to shared resources.

As for the examiner interpretation of the language of claim 22, namely that "the claim is understood the instruction identifies the signal by instructing the apparatus to determine whether the proper signal was received acknowledging receipt of the request or indicating the thread was to be switched or not" (Office Action, paragraph 19, pages 5-6), applicant first notes that it is not clear where the elements referred to by the examiner appear in Parady (e.g., where does Parady describe "instructing the apparatus to determine whether the proper signal was received"?). In any event, even assuming, *arguendo*, that the elements relied upon by the examiner in relation to the claim 22 rejection appear in Parady, applicant contends that none of "instructing the apparatus to determine whether the proper signal was received", "acknowledging receipt of the request," or "indicating the thread was to be switched or not", is the same as identifying in the instruction which signal, once generate, will cause a swapping execution to the thread associated with the instruction that identified the signal, as claim 22 requires.

Thus, Parady does not disclose or suggest "wherein the at least one instruction identifies the signal," as recited in applicant's claim 22.

It is believed that all the rejections and/or objections raised by the examiner have been addressed.

Canceled claims, if any, have been canceled without prejudice or disclaimer. Any circumstance in which the applicant has (a) addressed certain comments of the examiner does not mean that the applicant concedes other comments of the examiner, (b) made arguments for the patentability of some claims does not mean that there are not other good reasons for patentability of those claims and other claims, or (c) amended or canceled a claim does not mean that the applicant concedes any of the examiner's positions with respect to that claim or other claims.

Attorney's Docket No.: 10559-076002 / P7568C

Applicant : Debra Bernstein et al.
Serial No. : 10/643,438
Filed : August 19, 2003
Page : 12 of 12

Please apply any charges or credits to deposit account 06-1050, referencing attorney docket 10559-076002.

Respectfully submitted,

Date: February 17, 2006

Ido Rabinovitch
Reg. No. L0080
Attorney for Intel Corporation

PTO Customer No. 20985
Fish & Richardson P.C.
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

21269479.doc